Analyse de la consommation d'eau

Sujet n°10 — Corrigé

Question 1 *Consommation totale par jour*

```
def total_conso(donnees, jour):
    total = 0
    trouve = False
    for m in donnees:
        if m["jour"] == jour:
            total = total + m["chaude"] + m["froide"]
            trouve = True
    if not trouve:
        return None
    return total
```

Données fournies : `total_conso(donnees, "2025-02-04")` → `33` ; `"2025-02-05"` → `6` ; `"2025-12-25"` → `None`.

Question 2 *Détection de fuite nocturne*

```
def fuite_possible(donnees, jour):
    consecutifs = 0
    for m in donnees:
        if m["jour"] == jour and m["heure"] <= "05:00":
            if m["chaude"] + m["froide"] > 0:
                consecutifs = consecutifs + 1
                if consecutifs >= 3:
                    return True
            else:
                consecutifs = 0
    return False
```

Données fournies : `"2025-02-04"` → `False` (max 2 consécutifs non nuls) ; `"2025-02-05"` → `True` (trois consécutifs à partir de `01:00`).

Question 3 *Lissage : diagnostic et correction*

Cause : le cas intermédiaire divise par `2` au lieu de `3` (somme de 3 valeurs). **Correction** : remplacer `/ 2` par `/ 3` à cette seule ligne.

```
def lissage_conso(valeurs):
    lisse = []
    for i in range(len(valeurs)):
        if i == 0:
            m = (valeurs[i] + valeurs[i+1]) / 2
        elif i == len(valeurs) - 1:
            m = (valeurs[i-1] + valeurs[i]) / 2
        else:
            m = (valeurs[i-1] + valeurs[i] + valeurs[i+1]) / 3
        lisse.append(m)
    return lisse
```

`lissage_conso([10, 20, 30, 40, 50])` → `[15.0, 20.0, 30.0, 40.0, 45.0]`.

Question 4 *Lissage : cas limite non traité*

Cause : liste à un seul élément — la branche `i == 0` accède à `valeurs[1]` inexistant → `IndexError: list index out of range`. (La liste vide est correctement gérée.) **Correction** : garde au début de la fonction.

```
def lissage_conso(valeurs):
    if len(valeurs) <= 1:
        return list(valeurs)
    lisse = []
    for i in range(len(valeurs)):
        if i == 0:
            m = (valeurs[i] + valeurs[i+1]) / 2
        elif i == len(valeurs) - 1:
            m = (valeurs[i-1] + valeurs[i]) / 2
        else:
            m = (valeurs[i-1] + valeurs[i] + valeurs[i+1]) / 3
        lisse.append(m)
    return lisse
```